MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2015/2016

ECE4246 – ADVANCED OBJECT-ORIENTED DESIGN WITH JAVA

(All Sections / Groups)

16 OCTOBER 2015
9:00 A.M – 11:00 A.M.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This examination paper consists of FIVE (5) pages including cover with THREE (3) questions only.
2. Answer ALL questions. All questions carry 25 marks and the distribution of the marks for each question is given.
3. This examination is OPEN-BOOK. Allowed material: Textbook and/or bound notes only.
4. Please print all your answers in the Answer Booklet provided.

Question 1

Create a program which converts between ounce and gram (both ways).

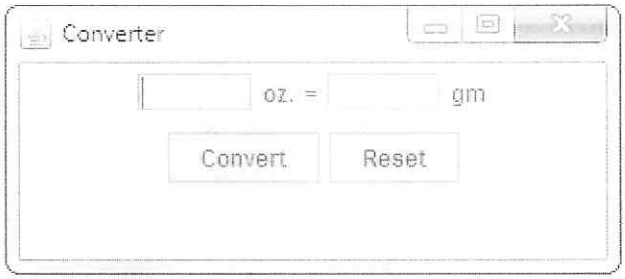

Figure Q1a

Write a class, **`ConverterPanel`**, which inherits from **`JPanel`**. Create, instantiate and set up GUI components shown in Figure Q1a. Use **`JTextField`** for inputs and **`JLabel`** to show the descriptions. Include buttons to convert and reset the fields.

[10 marks]

The convert button performs conversion of one unit to the other when only **ONE** of the fields has input. I.e., with input in "oz" field and blank "gm" field (Figure Q1b), the input is converted from ounce to gram with the result displayed in the "gm" field (Figure Q1b). Vice-versa when there is input in "gm" field and "oz" field is left blank. Both fields should be reset to blanks if the user clicks on the convert button with input values in both fields. The reset button resets both fields to blanks.

[10 marks]

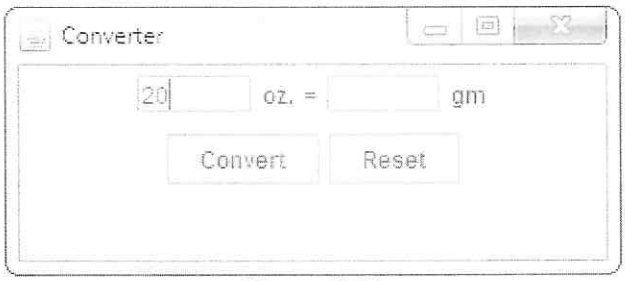

Figure Q1b

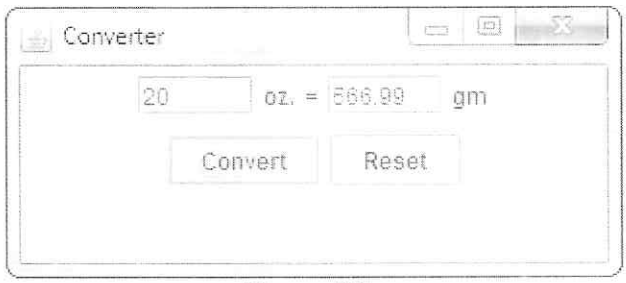

Figure Q1c

Continued...

Write an application program to display a **`ConverterPanel`** in a frame.

[5 marks]

NOTE: 1 ounce = 28.3495 grams

Question 2

Figure Q2a shows the UML class diagram for an abstract class, **`GeometricShape`**.

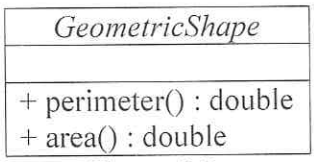

Figure Q2a

Write an interface, **`Graphical`**, with a method

```
draw(Graphics g, int upperX, int upperY)
```

[2 marks]

Class **`GeometricShape`** implements the interface **`Graphical`**. Write class **`GeometricShape`**.

[2 marks]

Write a class, **`Circle`**, which inherits from **`GeometricShape`**. Include attribute(s) that define a circle and implement all necessary methods, including accessors and mutators.

[9 marks]

Write an application class, **`Ring`**, which displays a circle in a frame, using **`JPanel`** and **`Circle`** objects. The circle spans 200 pixels across. See Figure Q2b.

[6 marks]

Draw the UML class diagrams to show the relationships among **`GeometricShape`**, **`Graphical`**, **`Circle`**, and **`Ring`**.

[6 marks]

Continued...

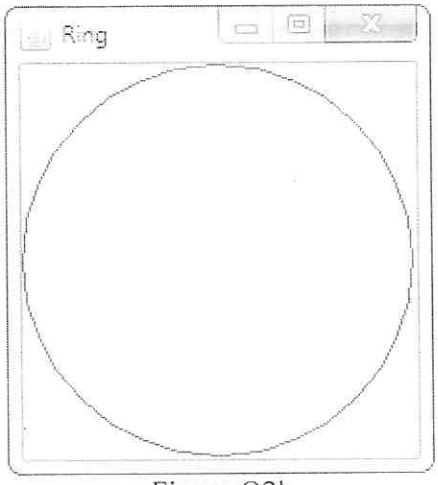

Figure Q2b

Hint: A frame is used to display a panel, which displays the rings. The panel object maybe implemented as an anonymous class within **`Ring`**, as a separate class, or otherwise.

Question 3

A company employs administrators and salespersons. An administrator draws a monthly basic salary whereas a salesperson earns a 5% commission on his / her monthly sales on top of a basic salary. The company needs a program to print a salary statement for all its employees. Figure Q3a shows a sample statement for administrators and Figure Q3b shows a sample statement for salespersons.

```
Salary Statement

Employee Name : Amy
Basic: $3500.0
Total: $3500.0
```

Figure Q3a

```
Salary Statement

Employee Name : John
Basic: $2000.0
Commission: $2250.0
Total: $4250.0
```

Figure Q3b

Continued...

Classes **Administrator** and **Salesperson** represent the two types of employees of the company. Both classes require attributes **name** and **basicSalary**, whereas **Salesperson** also has **salesAmount**.

To improve code efficiency and reusability, factor out the similar attributes and behavior of **Administrator** and **Salesperson** and implement them in an abstract superclass, **Employee**. **Employee** must be designed such that all its subclasses are required to implement their own versions of the methods to compute monthly salary and create salary statement:

```
public double computeNettSalary()
public String salaryStatement()
```

Write class **Employee**.

[6 marks]

Write class **Administrator**.

[5 marks]

Write class **Salesperson**.

[9 marks]

State ONE (1) pattern used in the design of the classes **Employee**, **Administrator** and **Salesperson**. Explain the context of the pattern and how the solution applies to the requirements of this question.

[5 marks]

NOTE: DO NOT write any application or driver classes.

End of Paper